Toward the Development of a High Fault Tolerant Learning System Based on the ALM Architecture

Masayuki Murakami
Department of Systems Engineering
University of Electro-Communications
Chofu 182–8585, Japan
masayuki@fs.se.uec.ac.jp

Nakaji Honda
Department of Systems Engineering
University of Electro-Communications
Chofu 182–8585, Japan
honda@se.uec.ac.jp

Abstract— **The active learning method (ALM), a methodology of soft computing, has been proposed as a new fuzzy-based modeling approach. The ALM has processing engines named IDS's, which are tasked with extracting useful information from a system subject to modeling. Hardware that implements the ALM provides fault tolerant capabilities due to parallelized IDS units. This paper discusses hardware redundancy of the ALM and traditional computing systems, and presents fault tolerant capabilities of the ALM architecture.**

I. Introduction

Soft computing is a discipline which features a low-cost, robust computing process in the presence of ambiguity and uncertainty, focusing on the remarkable human ability to effortlessly deal with intricate information. Thus, some soft computing methodologies simulate human behavior and physical characteristics. The ALM (Active Learning Method), which has been proposed as a methodology of soft computing, is a learning system that simulates the intelligent information handling processes of the human brain with a microscopic view [1][2][3]. The ALM is formed on the basis of fuzzy concepts, striving to achieve human-like information processing. Typical fuzzy applications use linguistic processing with membership functions of the kind seen in fuzzy rules. On the other hand, the ALM features pattern-based processing. This simulates human nature in the modeling of processing information in pattern-like images rather than utilizing numerical interpretations, when humans acquire knowledge from complex targets. In the ALM, a modeling method named IDS has been proposed. The ALM has plural IDS units, which are modeling processing engines where useful pattern information is generated directly from raw input/output data.

ALM systems should have a hardware architecture that enables parallel processing in the modeling layer which consists of plural IDS units. Each IDS unit perform pattern processing which is time-consuming in software calculation using a standard CPU. If the IDS is implemented in dedicated hardware, parallelization with the IDS hardware units not only boosts the overall processing speed, but also provides system fault tolerance due to hardware redundancy. Thus, it is desirable in terms of processing nature, performance, and robustness to develop dedicated hardware for the IDS. Reference [4] and [5] describe the development of IDS hardware and its effectiveness.

System fault tolerance is indispensable for industrial applications such as safety-critical or long-life systems. If those fault tolerant systems are utilized for modeling, inference, and control in the presence of ambiguity or uncertainty, exact computation, as seen in traditional computing systems, is not necessarily required. However, almost 100% of such systems are still based on the traditional computing architecture. Here, 'traditional' means the von Neumann architecture. Redundancy is a common means to enhance fault tolerance of systems. Hardware redundancy is the most effective for improving system availability. For example, compared with a single CPU unit system, a duplex system with hot-swap capable CPU units dramatically increases system availability. Design of fault tolerant systems based on the traditional computing architecture is difficult due to its exact and sequential processing nature, and fault tolerance of those systems cannot be perfect, because there is a shared area in the redundant architecture.

The processing of ALM is inexact and parallel in nature. Thus, hardware redundancy of the ALM is simply constructed, and it has the possibility of overcoming many drawbacks of the traditional fault tolerant systems. In this paper, we discuss hardware redundancy of the ALM and the traditional computing systems, and show fault tolerant capabilities of the ALM through software-based simulated failure tests.

II. Active Learning Method

A. Fundamentals of the ALM

The ALM results from algorithm modeling focused on simulating the intelligent information handling processes of the human brain based on the following set of hypotheses.

When humans engage in modeling a complex system:

1 They derive system features by breaking down the system into simpler aspects and translating information in a more readily comprehensible form.

2 The information obtained at this stage is of a general outline nature, representing pattern-like images rather than numerical data.

3 They link multiple images together, and strive to obtain an understanding of the system as a whole.

4 If the information is inadequate, an effort is made to acquire additional information from specific parts of the system by active manipulation. This process is repeated through trial-and-error.

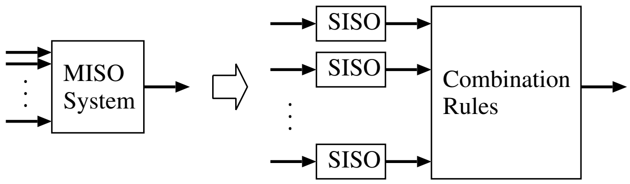


Fig. 1. The modeling approach of ALM

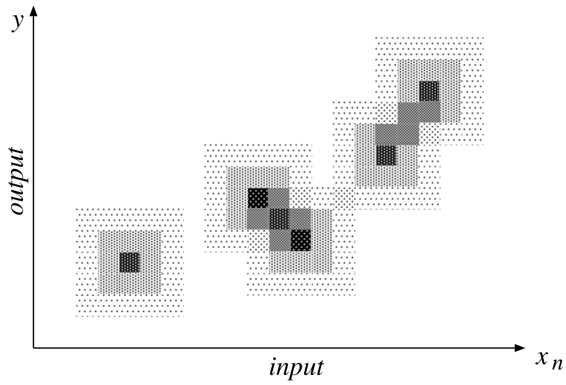


Fig. 2. Image of data spread in IDS

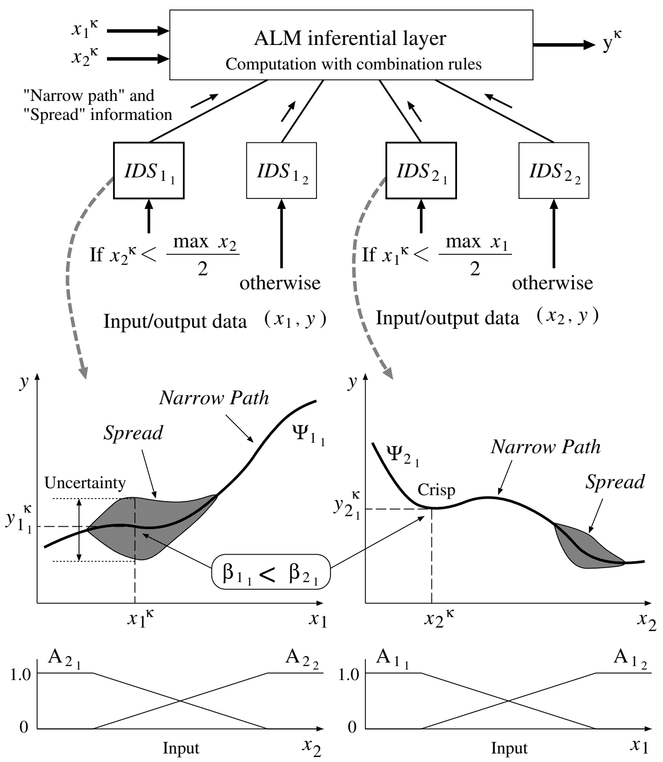


Fig. 3. Inference in the ALM using narrow paths and spreads

Fig. 1 shows the modeling approach of ALM. Step 1 corresponds to the breakdown of a multi-input, single-output system (MISO), $y = f(x_1, ..., x_N)$, into single-input(x_n), single-output(y) systems (SISOs). From input/output data, each SISO makes a pattern useful for modeling, using the method of ink drop spread (IDS). Pattern information is obtained through data fuzzification, and is visually comprehensible. The general concept and algorithm for IDS are explained below.

IDS is an SISO system that contains an x_n-y plane for the recording of input/output relationships. The method plots input/output data on this x_n-y plane, and blurs data points like ink patterns, as illustrated in Fig. 2. This process is called "data spread." As individual data spreads overlap, the overlapping portions become increasingly darker, ultimately resulting in a pattern on the surface of the plane. The pattern includes continuous line(s) and spread(s). The continuous line shows that the input and output have a close relationship, and the spread indicates that the output depends upon other input(s). In particular, the continuous line is referred to as a 'narrow path.' Practically, the narrow path and the spread are calculated in a manner that determines the mean and the dispersion. The ALM architecturally consists of a modeling layer with plural IDS units and an inferential layer. The narrow paths and spreads are transferred from the IDS units to the upper inferential layer, and are used for the inferential process in the ALM.

The pattern sets thus obtained are next combined on the basis of a set of combination rules. For example, with a two-input system comprising x_1 and x_2, the output, y, is sought with the following combination rules when each input is divided into two regions.

$$\begin{aligned} &R_1 : if\ x_2\ is\ A_{2_1},\ then\ y_{1_1}\ is\ \Psi_{1_1} \\ &R_2 : if\ x_2\ is\ A_{2_2},\ then\ y_{1_2}\ is\ \Psi_{1_2} \\ &R_3 : if\ x_1\ is\ A_{1_1},\ then\ y_{2_1}\ is\ \Psi_{2_1} \\ &R_4 : if\ x_1\ is\ A_{1_2},\ then\ y_{2_2}\ is\ \Psi_{2_2} \\ &y\ is\ \beta_{1_1} y_{1_1}\ or\ \beta_{1_2} y_{1_2}\ or\ \beta_{2_1} y_{2_1}\ or\ \beta_{2_2} y_{2_2} \end{aligned} \tag{1}$$

where A_{n_m} is the membership function, expressing the mth region for the nth input variable. Ψ_{n_k} represents the kth narrow path for the nth input variable. β_{n_k} is the degree of confidence for each narrow path, and is a weighted average when the narrow paths are combined. The value of β_{n_k} is determined by the size of the spread. Fig. 3 shows an example of the pattern-based modeling process on the two-input system described above. In this figure, the ALM system seeks an output y^κ for x_1^κ and x_2^κ. The two graphs of the x_n-y plane, including narrow paths and spreads, indicate that we can obtain an accurate y^κ with pattern information from IDS_{2_1} rather than from IDS_{1_1}. In spite of the simplified modeling process, the ALM is able to output excellent modeling results using plural IDS units.

B. The IDS Method

An IDS unit has a processing controller and an x_n-y plane. As the main function of IDS, the controller executes the data spread on the plane with input/output data sets, and calculates the narrow path and the spread from the pattern image on the plane. Let the x_n-y plane be $P_{x_n y} = \{p(x, y) | x \in X_n, y \in Y\}$ where $p(x, y)$ is the point (x, y) in the plane. $d(x, y)$ denotes the darkness of the point (x, y).

If a data spread is done at (x_s, y_s), darkness of the neighborhood of (x_s, y_s) is increased. As a pattern of ink drops, we defined this added darkness by the following expression,

$$\varDelta d(x_s+u, y_s+v) = I_0 \exp(-I_{bias}(u^2+v^2)) \quad (2)$$
$$-I_s \le u, v \le I_s$$

where I_0, $I_{bias} > 0$, and I_s are parameters for the added darkness to the center point of ink drops, the bias of darkness according to the distance from the center point, and the spread of ink drops, respectively.

The important factors of IDS pattern image are narrow path and spread. We used simplified methods of determining the narrow path and spread so that utilized hardware resources and processing time could be reduced. We applied the bisector of area method (BOA) and a simplified method named IDS α-cut spread to calculations of the narrow path and the spread, respectively. In fuzzy applications, the BOA is often used in defuzzification. The narrow path using the BOA is calculated by

$$\psi^{BOA}(x) = \{b | \sum_{y=1}^{b} d(x,y) \approx \sum_{y=b}^{y_{max}} d(x,y),\ b \in Y\} \quad (3)$$

where $y_{max} = \max_{y \in Y} y$. The IDS α-cut spread is defined by

$$\sigma_\alpha^{IDS}(x) = \max_{y \in Y}\{y | d(x,y) > \alpha\} - \min_{y \in Y}\{y | d(x,y) > \alpha\}. \quad (4)$$

The IDS α-cut spread is based on the idea behind α-cut of fuzzy sets. The value of α may be determined by an average darkness of the whole x_n-y plane.

Spread information in the x_n-y plane is used to seek the degree of confidence of the narrow path for the inferential processes of the ALM. The degree of confidence of the narrow path is obtained from spread $\sigma_\alpha^{IDS}(x)$, using a conversion function,

$$f_c(x) = \begin{cases} 1, & if \quad x \le P \\ exp(-C_{bias}\sigma_\alpha^{IDS}(x)), & if \quad P < x \end{cases} \quad (5)$$

where $C_{bias} > 0$ is a parameter for bias, and f_c also has a function of normalization.

III. IDS Hardware

In ALM, the number of narrow paths and spreads sought with IDS increases extremely as the number of inputs and input divisions increases. A system has N inputs, and each input X_1, ..., X_N is divided to M_1, ..., M_N regions, respectively. The total number of IDS units, U_{total} is given by

$$U_{total} = \sum_{i=1}^{N} \prod_{j=1, j \ne i}^{N} M_j. \quad (6)$$

Suppose we have an ALM system with a number of IDS units integrated into a standard computer. Executing the data spread and calculating the narrow path and spread for each IDS unit would place a heavy load on a host CPU when a complex system is being modeled. This would lengthen the processing time required for modeling and impact system performance on the whole.

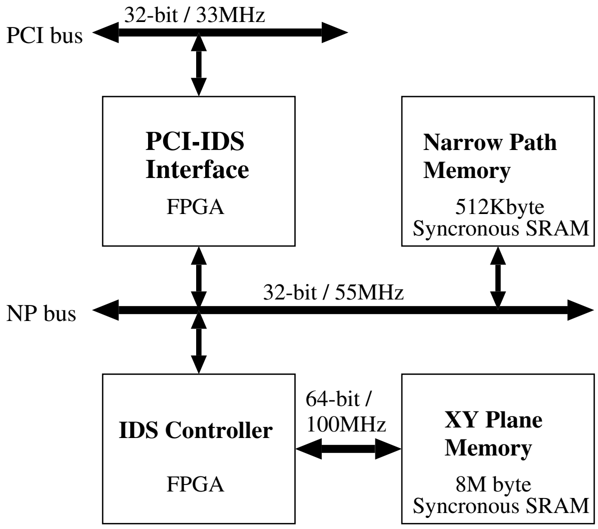


Fig. 4. Block Scheme of the HIDS

The ALM system structure provides robustness against failures. An IDS unit executes a given modeling process independently from other IDS units, but partially supplements the results from these other IDS units. In Fig. 3, IDS_{1_1} supplements IDS_{2_1}. In case of a breakdown in the IDS_{2_1} unit, the system may keep a certain level of output accuracy. The IDS layer conducts most parts of modeling process and transfer condensed pattern data to the upper inferential layer. The amount of data transferred between the upper layer unit and each IDS unit is small. These conditions also facilitate system scalability.

In order to secure the performance and fault tolerance of the ALM, dedicated hardware for IDS is indispensable for complex modeling which demands realtime capabilities, and industrial applications which demand high system availability.

According to the hardware implementations described in Section II, we developed a new IDS hardware unit named 'HIDS'. The block scheme of the HIDS is illustrated in Fig. 4. The main IDS circuits ran at 100MHz. The NP bus in the figure is a simple original bus. The HIDS is installed as a 32-bit/33MHz standard PCI card. Reference [5] gives details of the development of the HIDS.

IV. Hardware Redundancy

A. Redundancy of computing systems

Most types of traditional computers run with a single processor, and a single unit which is comprised of a processor, a memory controller, the main memory, and other peripherals. Although such computers provide us with exact computational results, if we demand severe fault tolerance of the computers, the control logic for redundancy becomes very complex because of the exact and sequential nature of the traditional computing. Fig. 5 illustrates a typical example of 'pair and a spare' which is the most well-established fault tolerant technique in traditional computing systems, and is widely used in safety-critical applications that require high system availability. The central controller in Fig. 5 is a memory controller, and also has the functions of error detection for

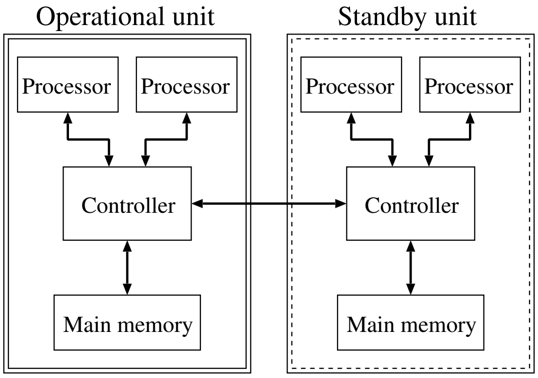


Fig. 5. Redundancy of traditional computers using 'pair and a spare'

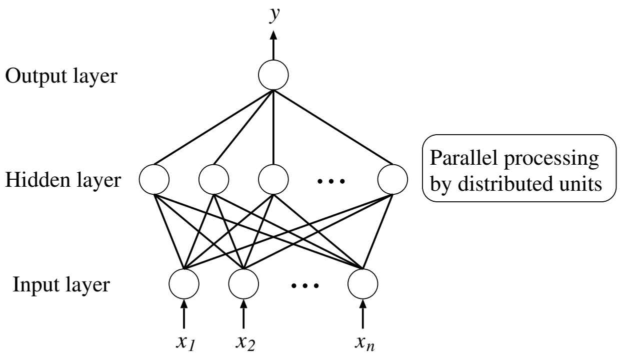


Fig. 6. A neural network structure

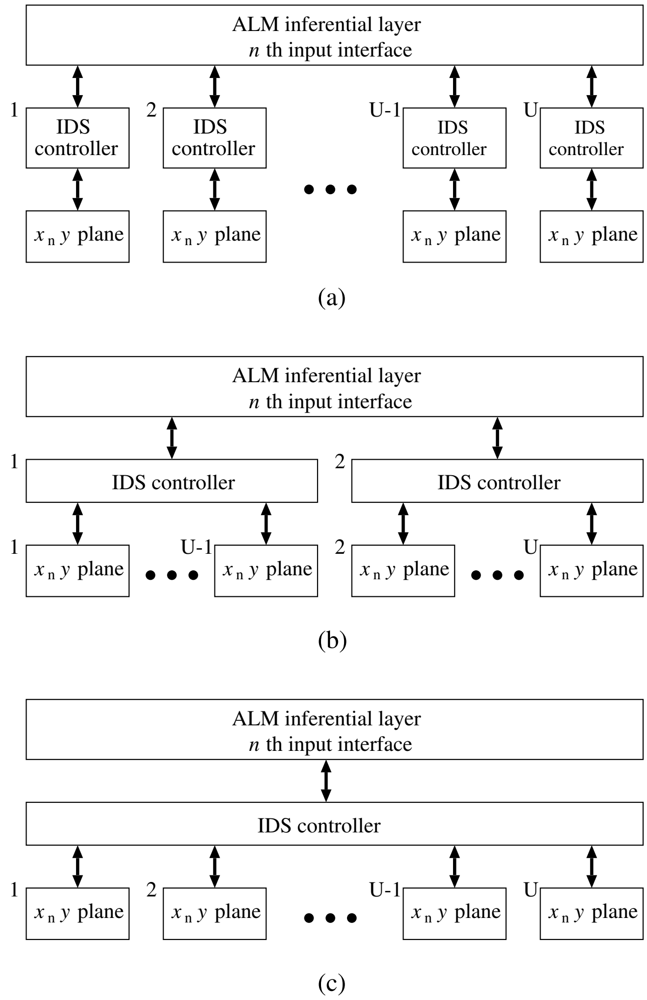


Fig. 7. ALM modeling layer structures

duplicate processors and bus arbitration between redundant units. If duplicate processors output different computational results in an operational unit, the faulty unit is removed from the operation and immediately replaced with a standby unit. In the traditional fault tolerant systems, duplicate processors do not contribute to the enhancement of the processing speed of the unit. On the contrary the overall speed may be degraded due to control for redundancy in comparison with a standard architecture. Also, the fault tolerance of traditional computers cannot be perfect, because there is a shared area in the redundant architecture. As this shared area decreases, the availability of the redundant system improves.

Unlike the traditional computing systems above, neural networks do not provide exact computational results. They are applied mainly to control and learning problems, and their inexact processing nature is a feature of soft computing systems. Although, in the hardware of soft computing technologies, neural network hardware [6] may have the most potential to be used in a wide variety of industrial applications, the industrial use of neural networks in hardware is still limited. Fig. 6 illustrates a neural network structure. The neural network systems have massively interconnected structures and perform heavy parallel processing if dedicated hardware is used for them. Hence, they have considerable potential for fault tolerance. The fault tolerance of neural networks has attracted the attention of many researchers for years [7][8][9].

B. Redundancy of ALM systems

The ALM system, as also seen in neural networks, has 'natural' fault tolerance in its architecture. For modeling complex systems, the ALM involves a good number of IDS units, each of which has different modeling tasks. Parallelization with IDS hardware units increases not only the overall processing speed and modeling accuracy, but also system fault tolerance. Fig. 7 shows three structures of the ALM modeling layer. This figure illustrates part of the modeling layer which deals with the nth input X_n. Fig. 7(a) represents the highest redundancy, where IDS controllers are equipped with the total number of divisions of the other input variables as shown in (6). In this structure, the failure of IDS controllers has the least impact on the precision of the overall model. For large-scale systems, however, if hardware implementation of the IDS uses digital circuitry, this structure would be expensive because a great number of IDS controllers would be required. Fig. 7(b) and 7(c) are alternative structures that are not expensive. In particular, Fig. 7(c) is the least expensive structure among ALM systems that have realtime capabilities. If some intervals between input/output data sets are given, and these are not simultaneously input into the same nth input interface, one IDS controller can execute modeling processing in realtime.

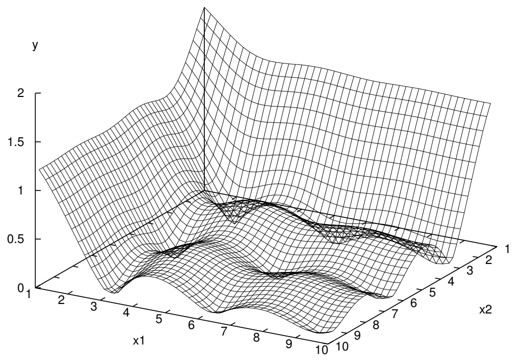


Fig. 8. A graph of function (7)

TABLE I
IDS PARAMETERS IN EXPERIMENTS

I_0	I_{bias}	I_s	P	C_{bias}	α
15	0.0083	39	90	0.01	1

However, failure of the IDS controller in this structure results in the lack of a modeling means for the aspect of input X_n. Hardware redundancy of the ALM modeling layer would be determined according to the system scale, system reliability, and cost.

C. Experiments on fault tolerance of the ALM

To examine modeling performance in the presence of a single unit failure in the Fig. 7(a) structure, we conducted software-based simulated failure tests. We gave a fault to each of IDS units in turn, and kept the upper inferential unit from receiving any data from the faulty IDS unit. In the tests, each IDS unit had an x_n-y plane with 1024x1024 resolution, and IDS parameters were set as Table I. The following nonlinear function was used for modeling.

$$y = \sqrt{2\Big(\frac{\sin x_1}{x_1}\Big)^2 + 3\Big(\frac{\sin x_2}{x_2}\Big)^2}, \quad 1 \leq x_1, x_2 \leq 10 \qquad (7)$$

This function is graphically shown in Fig. 8. We tried three different modeling conditions in Table II. Each input was divided into two, four, and six regions. By changing the number of input divisions, we can see the relationship between the redundancy of IDS units and the fault tolerance of ALM.

To evaluate the completed models, 1000 input sets that were randomly generated were input to the models, and the mean squared error (MSE) and the correlation coefficient were calculated using the original function's outputs and the model outputs. Fig. 9, Fig. 10, and Fig. 11 map the results of modeling tests that divided each input by two, four, and six, respectively. Each case (a) in the figures is a modeling result without any unit failure, and each case (b) represents the worst modeling result in MSE among the ALM systems with a single unit failure. In Fig. 9(b) part of the graph was apparently out of shape, because one out of four IDS units did not work. In Fig. 11(a) the modeling result of six divisions in no failure was very similar to the original, and in Fig. 11(b) the effect of a single unit failure remarkably reduced. As a result of the experiments, we confirmed that if more IDS units were used according to the input divisions, the fault tolerance of the ALM system improved.

TABLE II
MODELING CONDITIONS IN EXPERIMENTS

	ALM1	ALM2	ALM3
Number of input divisions	2	4	6
Number of IDS units	4	8	12
Number of training data	200	350	500

V. CONCLUSION

Today's fault tolerant systems have several issues to be discussed, as described in section IV.*A*. Those issues have been caused by excessive dependence on the power and technology of modern-day computers based on the von Neumann architecture. We consider that soft computing technologies are great potential for the development in the area of fault tolerant systems. We aim to realize an ALM learning system that is capable of performing high-speed high-precision computation with excellent fault tolerant capabilities. In this paper, the redundancy of the ALM architecture and its fault tolerant capabilities were presented. The hardware redundancy of the ALM contributes not only to the processing speed and modeling accuracy, but also to system fault tolerance. In future papers, we will compare the ALM with neural networks about various performance criteria including fault tolerant capabilities.

REFERENCES

[1] S. B. Shouraki, "A Novel Fuzzy Approach to Modeling and Control and its Hardware Implementation Based on Brain Functionality and Specifications," Ph.D. dissertation, The Univ. of Electro-Communications, Chofu, Japan, March 2000.

[2] S. B. Shouraki and N. Honda, "Recursive Fuzzy Modeling Based on Fuzzy Interpolation," *Journal of Advanced Computational Intelligence*, Vol. 3, No. 2, pp. 114 – 125, April 1999.

[3] S. B. Shouraki, G. Yuasa, and N. Honda, "Fuzzy Interpretation of Human Intelligence," *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, Vol. 7, No. 4, pp. 407 – 414, August 1999.

[4] M. Murakami, N. Honda, and J. Nishino, "A Hardware Design for a New Learning Systems Based on Fuzzy Concepts," *Proceedings of the 12th IEEE International Conference on Fuzzy Systems*, May 2003, pp. 1002 – 1007.

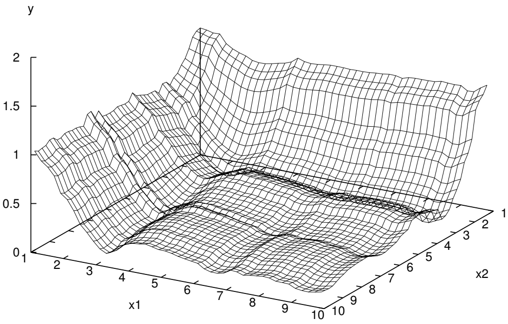
[5] M. Murakami, N. Honda, and J. Nishino, "A High Performance IDS Processing Unit for New Fuzzy-based Modeling," *Proceedings of the 13th IEEE International Conference on Fuzzy Systems*, July 2004, pp. 935 – 940.

[6] T. Schoenauer, A. Jahnke, U. Roth, and H. Klar, "Digital Neurohardware: Principles and Perspectives," *Proceedings of the 3rd International Workshop on Neural Networks in Applications*, February 1998, pp. 101 – 106.

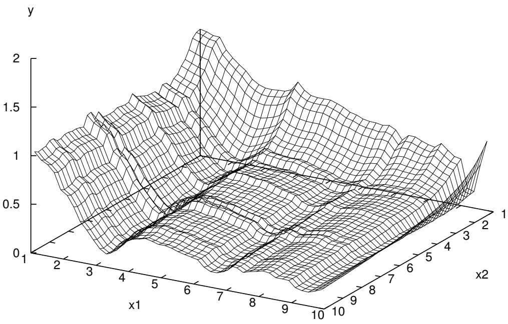
[7] G. Bolt, "Assessing the Reliability of Artificial Neural Networks," *Proceedings of the 1991 IEEE International Joint Conference on Neural Networks*, November 1991, pp. 578 – 583.

[8] J. Schumann, P. Gupta and S. Nelson, "On Verification & Validation of Neural Network Based Controllers," *Proceedings of the 8th International Conference on Engineering Application of Neural Networks*, September 2003.

[9] L. Vintenat, "Are Neural Networks a Fault-tolerant Solution for Space Missions ?" *Proceedings of DASIA'98 – Data Systems in Aerospace*, May 1998.

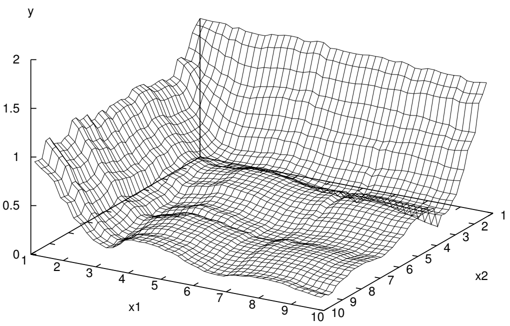


(a) No failure, MSE = 0.0118, correlation coefficient = 0.991

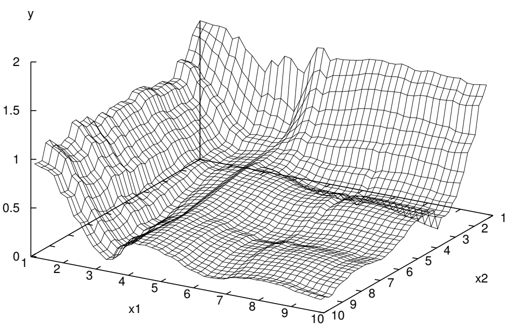


(b) Single failure, MSE = 0.0558, correlation coefficient = 0.850

Fig. 9. Model outputs of ALM1 (two divisions)

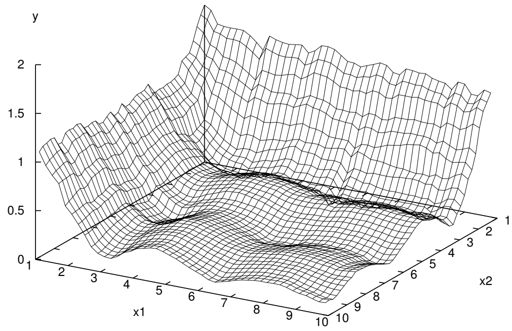


(a) No failure, MSE = 0.0068, correlation coefficient = 0.989

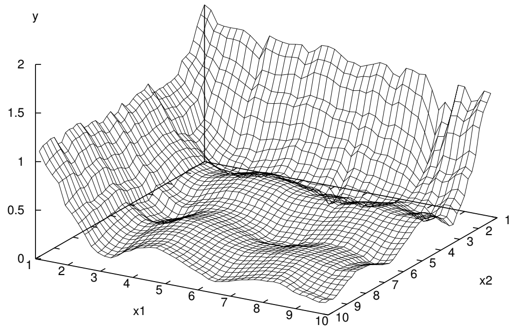


(b) Single failure, MSE = 0.0129, correlation coefficient = 0.967

Fig. 10. Model outputs of ALM2 (four divisions)



(a) No failure, MSE = 0.0065, correlation coefficient = 0.993



(b) Single failure, MSE = 0.0102, correlation coefficient = 0.985

Fig. 11. Model outputs of ALM3 (six divisions)